4-22-92
E 6853

NASA Technical Memorandum 105432

Interactive Solution-Adaptive Grid Generation Procedure

Todd L. Henderson
University of Illinois
Urbana, Illinois

Yung K. Choo
Lewis Research Center
Cleveland, Ohio

and

Ki D. Lee
University of Illinois
Urbana, Illinois

April 1992

INTERACTIVE SOLUTION-ADAPTIVE GRID GENERATION PROCEDURE

Todd L. Henderson*
University of Illinois
Urbana, Illinois 61801

Yung K. Choo
Lewis Research Center
Cleveland, Ohio 44135

and

Ki D. Lee
University of Illinois
Urbana, Illinois 61801

E-6853

Summary

TURBO-AD is an interactive solution-adaptive grid generation program under development. The program combines an interactive algebraic grid generation technique and a solution-adaptive grid generation technique into a single interactive solution-adaptive grid generation package. The control point form uses a sparse collection of control points to algebraically generate a field grid. This technique provides local grid control capability and is well suited to interactive work due to its speed and efficiency. A mapping from the physical domain to a parametric domain was used to improve difficulties that had been encountered near outwardly concave boundaries in the control point technique. Therefore, all grid modifications are performed on a unit square in the parametric domain, and the new adapted grid in the parametric domain is then mapped back to the physical domain. The grid adaptation is achieved by first adapting the control points to a numerical solution in the parametric domain using control sources obtained from flow properties. Then a new modified grid is generated from the adapted control net. This solution-adaptive grid generation process is efficient because the number of control points is much less than the number of grid points and the generation of a new grid from the adapted control net is an efficient algebraic process. TURBO-AD provides the user with both local and global grid controls.

*Summer Student Intern at NASA Lewis Research Center.

1. Introduction

Quality grids are needed to obtain accurate numerical simulations of complex flows. But generating quality grids is difficult for complex problems because the grids must be provided without a prior knowledge of the flow details. Precise local and global grid controls are often difficult even when details of the flow are known. A grid adaptation process (refs. 1 to 3) that couples the grid with the evolving flow solution can provide a means of obtaining proper distribution of grid points. This adaptation process is combined with the control point form of algebraic grid generation (refs. 4 to 6) to improve the efficiency of the adaptation process and to provide explicit local/global grid control capability.

TURBO-AD under development is a 2D code. It starts with a given initial grid and a given initial solution, and it redistributes the grid points to provide dense grid in high gradient areas of a solution while having sparse grid in low gradient areas. The new flow solution obtained from the adapted grid is able to capture more significant flow features in the high gradient areas. The program runs on the IRIS 4D series workstations interactively. The interactive process of the code allows the user to immediately see the results and modify the grid as desired along the way.

In TURBO-AD, all grid modifications are performed in a parametric domain. The grid adaptation is obtained through adapting the control points to a source distribution, which is defined by the gradients of a numerical flow solution. The new adapted grid is then computed from the new adapted control points. The control point adaptation offers advantages over the direct grid adaptation. It increases the efficiency of the solution-adaptive procedure because the number of control points is significantly less than the number of grid points. It also provides smooth grids resulting from the control point form. The solution adaptation of a grid does not always result in a smooth grid for many adaptation techniques. But, when a nonsmooth set of control points is used to generate a grid the resulting grid is much smoother. The grid modification in the square parametric domain offers another advantage. It allows the control points and grid points on the boundary to freely move with very little alteration of the given boundary geometry. It also prevents grids from tangling, which can happen in concave regions with the control point formulation since there are no concave boundaries in the parametric domain.

This report discusses the technologies as they are applied to TURBO-AD and demonstrates TURBO-AD's capabilities through an example. A flow chart of the example and the input and output file descriptions are included in appendixes A and B, respectively.

2. Algebraic Control Point Grid Generation

The control point array is a sparse grid-type arrangement of locations in space with an index for each direction. In two dimensions, it will be denoted by $(C_{i,j})$. As an algebraic method, the control point form provides explicit control of the grid shape and spacing through the dynamic movement of the control points. Figure 1(a) shows an example of a two-dimensional control point array.

A fundamental part of the control point formulation is the construction of curves. This construction represents algebraic coordinate generation in a single direction between two opposing boundaries (fig. 1(b)). These boundaries are the fixed endpoints of the curves. The remaining control points are in the interior of the sequence and are used to control the shape of the curve.

From any one control point, the two neighboring points on each side form local segments within the entire piecewise linear curve. The local segments about each control point define a change in direction as shown in figure 1. The rate of change is determined by the two linear connections attached to the given control point as the curve assumes the respective directions between each pair of control points.

A continuous direction field is obtained in a smooth manner by interpolation. The independent variable for the interpolation is the curve parametrization. The interpolated result defines the field of vectors that are tangent to the desired curve. Simply stated, it is an interpolation of first parametric derivatives. This determines a smooth first derivative of the entire curve. The desired curve is then obtained by a parametric integration. The integration here is taken so that the curve connects the specified endpoints.

To state the result mathematically, let $C_{1,j}, C_{2,j}, \dots C_{M,j}$ be the given sequence of M points in space as shown in figure 1(b) (e.g., $C_{1,2}, C_{2,2}, \dots C_{5,2}$ for j=2 in fig. 1(a)), ξ be the curve parameterization, $E_j(\xi)$ be the position at ξ along the desired curve, $\xi_1, \xi_2, \dots \xi_{M-1}$ be the successive parametric locations used to interpolate the directions of $(C_{2,j}-C_{1,j})$, $(C_{3,j}-C_{2,j})$, ..., $(C_{M,j}-C_{M-1,j})$, and $\Psi_1, \Psi_2, \dots \Psi_{M-1}$ be the corresponding interpolation functions which separate each successive pair of segments by assuming a nonzero value at the associated location while vanishing at the remaining locations for interpolation. In two dimensions $C_{\alpha,j}=(x_{a,j},y_{a,j})$ and $E_j(\xi)=[x_j(\xi),y_j(\xi)]$. Using this notation, the desired curve is given by

$$E_j(\xi) = C_{1,j} + \sum_{\alpha=1}^{M-1} G_\alpha(\xi)[C_{\alpha+1,j} - C_{\alpha,j}] \qquad \text{for} \quad j = 1,2,\dots,N \tag{1}$$

where

$$G_\alpha(\xi) = \int_{\xi_1}^{\xi} \Psi_\alpha(\mu)d\mu \tag{2}$$

To obtain local grid controls, local interpolation functions are used. With local functions, the movement of a control point results in an alteration of the constructed curve that is restricted to a local region about the point. The remaining regions are unaltered. The simplest local interpolants are the piecewise linear functions that do not vanish over at most two intervals defined by $\xi_1 < \xi_2 < \dots < \xi_{M-1}$. With the uniform partition $\xi_k = k$ for simplicity in discussion, an explicit form of the normalized interpolants, which yields $G_k(\xi_{N-1}) = 1$ for all k, can be given by

$$\psi_1(\xi) = \begin{cases} 2(2-\xi) & \text{for} \quad 1 \le \xi < 2 \\ 0 & \text{for} \quad 2 \le \xi \le (M-1) \end{cases} \tag{3}$$

$$\psi_k(\xi) = \begin{cases} 0 & \text{for} \quad 1 \le \xi < (k - 1) \\ (\xi - k) + 1 & \text{for} \quad (k - 1) \le \xi < k \\ (k - \xi) + 1 & \text{for} \quad k \le \xi < (k + 1) \\ 0 & \text{for} \quad (k + 1) \le \xi \le (M - 1) \end{cases} \tag{4}$$

$$\psi_{N-1}(\xi) = \begin{cases} 0 & \text{for} \quad 1 \le \xi < (M - 2) \\ 2(\xi - M + 2) & \text{for} \quad (M - 2) \le \xi \le (M - 1) \end{cases} \tag{5}$$

Similarly, a curve can be constructed for the other direction for the index of i:

$$F_i(\eta) = C_{i,1} + \sum_{\beta=1}^{N-1} H_\beta(\eta)[C_{i,\beta+1} - C_{i,\beta}]$$

$$\text{for } i = 1,2,\ldots,M \quad (6)$$

where η is the curve parameterization and $H_b(\eta)$ the integration of the interpolants. The tensor product form depends only on $C_{i,j}$ and is given by

$$T(\xi,\eta) = E_1(\xi) + \sum_{\beta=1}^{N-1} H_\beta(\eta)[E_{\beta+1}(\xi) - E_\beta(\xi)] \quad (7)$$

or alternatively by

$$T(\xi,\eta) = F_1(\eta) + \sum_{\alpha=1}^{M-1} G_\alpha(\xi)[F_{\alpha+1}(\eta) - F_\alpha(\eta)] \quad (8)$$

These two expressions are equivalent. The tensor product matches E_j or F_i at the extremities of i and j (i.e., at corner points in fig. 1(a) for instance).

When boundaries are to be specified, the corresponding data appear at the extremities of the values for ξ and η. Since the coordinate transformations are generally expressed in the form of a vector for the desired positions of all points in physical space, it is convenient to express the boundary specifications in terms of the position vector. Thus, the boundaries are denoted by $P(1,\eta)$, $P(M-1,\eta)$, $P(\xi,1)$, and $P(\xi, N-1)$. To include the boundaries, the multisurface transformation is performed again as previously, but now with the actual boundaries inserted. This results in a modification of T for the ξ and η directions, respectively. In each such directional construction, the actual boundaries appear as end conditions for the corresponding variable while the remaining boundaries are generated by the control points. Thus, when T is subtracted from the sum of both directional constructions, the actual boundaries become end conditions for each variable. This process follows a Boolean sum format that upon simplification becomes

$$\begin{aligned} Q(\xi,\eta) = T(\xi,\eta) &+ \gamma_1[1 - G_1(\xi)][P(1,\eta) - F_1(\eta)] \\ &+ \gamma_2 G_{N-1}(\xi)[P(N-1,\eta) - F_N(\eta)] \\ &+ \gamma_3[1 - H_1(\eta)][P(\xi,1) - E_1(\xi)] \\ &+ \gamma_4 H_{M-1}(\eta)][P(\xi,M-1) - E_M(\xi)] \end{aligned} \quad (9)$$

where each of the four terms following the tensor product $T(\xi,\eta)$ represents a transfinite conformity to a boundary when each boundary on-off switch is 1. By setting any γ_i to 0, the corresponding boundary becomes available for free form modeling by means of the control points. In the order of appearance, the boundaries are for $\xi = 1$, $\xi = M - 1$, $\eta = 1$, and $\eta = N - 1$. Further details of the control point form are presented in references 4 to 6.

3. Parametric Mapping Technique

The mapping concept was implemented to alleviate two difficulties of the control point formulation near boundaries. The first problem is centered around the use of Dirichlet boundary conditions on outwardly concave boundaries. In concave regions it is possible for the second control net line to be outside the physical boundaries as shown in figure 2(a). This results in a tangled grid as shown in figure 2(b). The parametric domain uses a unit square which by definition has no concave boundaries. Therefore, the second control net line can not be outside the domain. The second problem arises when the boundary grid points are allowed to move using the free-form boundary condition. The free-form boundary condition allows the shape of the boundary to change when applied to a curve. This is not acceptable. However, if the boundary is straight, such as the parametric domain boundaries, the new grid points will fall on the boundary of the given geometry. If the grid points fall on the boundary in the parametric domain, the mapping back to the physical space insures that the points are on the boundary in the physical domain. This allows the grid points to float along the boundary without sacrificing the boundary shape.

The parametric domain is constructed such that it is a unit square with a grid point distribution which reflects the distribution in the physical domain. The s coordinates are constructed using the normalized arc length of one physical coordinate set of curves as shown in equation (10) and the t coordinates are likewise obtained using the other set of curves as shown in equation (11). An example of the grid and control net in both the physical domain and the resulting parametric domain can be seen in figure 3. If a particular boundary collapses to a single point in the physical domain, the parametric domain will be uniformly discretized for that boundary:

$$s_{i,j} = \frac{\sum_{\ell=2}^{i} \sqrt{(x_{\ell,j} - x_{\ell-1,j})^2 + (y_{\ell,j} - y_{\ell-1,j})^2}}{\sum_{\ell=2}^{i\max} \sqrt{(x_{\ell,j} - x_{\ell-1,j})^2 + (y_{\ell,j} - y_{\ell-1,j})^2}} \quad (10)$$

$$t_{i,j} = \frac{\sum_{\ell=2}^{j} \sqrt{(x_{i,\ell} - x_{i,\ell-1})^2 + (y_{i,\ell} - y_{i,\ell-1})^2}}{\sum_{\ell=2}^{j\max} \sqrt{(x_{i,\ell} - x_{i,\ell-1})^2 + (y_{i,\ell} - y_{i,\ell-1})^2}} \quad (11)$$

In TURBO-AD, the adaptation procedure takes place in the parametric domain as explained in section 4. Once the adaptation process is completed, the new adapted grid and control net must be inversely mapped to the physical space. This is accomplished

in three parts: the corner points, the boundaries, and the interior.

The corner points cannot be moved in the parametric space. Therefore, they can never be moved or relocated in the physical space. This makes the corner points trivial and they are merely the original points. The grid points on the boundaries in the parametric space will only move along the parametric boundary curve. Therefore, the boundary grid points are mapped from the parametric space to the physical space using a linear interpolation. This is a rather simple interpolation but is, in general, very reliable and robust. In the future more elaborate interpolations may be implemented such as the damped cubic spline (ref. 2). After the corner and boundary points are inversely mapped, the interior is inversely mapped from the parametric domain to the physical domain. The interior points are inversely mapped from the parametric domain to the physical domain using a bi-linear interpolation. This is a relatively quick and robust method to determine the new grid points in the physical domain. It is important to note that in all cases the initial physical and parametric points are used as the reference data for the interpolations. Therefore, the new boundary curves will not be deteriorated from repeated grid modifications more than a linear interpolation from the initial curves.

4. Solution-Adaptive Control Point Generation

The adaptation of grids to a solution is usually accomplished by directly adapting all the grid points. This can be time consuming for many adaptation algorithms. To increase the grid adaptation speed, the procedure described in reference 1 is used to adapt the control net instead of the grid. This new approach can easily be more than ten (10) times faster in two dimensions than the usual direct grid adaptation procedure of reference 1. The new grid generation from the adapted control net is an efficient algebraic process.

The adaptation procedure using the control points is illustrated in figure 4. It starts from the initial control net in the physical space (x,y) as shown in figure 4(a). The position of the initial control points is determined by the distribution of the initial grid points, from which the control net is constructed by "attachment" (ref. 5), and the definition of the local interpolation functions. As in equations (3) to (5), the interpolation functions about the first and final partition points are non-zero over one interval, while the interpolation functions over the internal partition points are non-zero over two intervals. Resulting from this is the first fine control net meshes around the boundary as seen in figure 4(a). The procedure then defines a control net in the parametric space (s,t) as shown in figure 4(b). This control net is mapped into another parametric space(u,v) to establish a uniform control net as shown in figure 4(c). The initial grid is also mapped into the parametric space (u,v) using the same procedure. Then two control sources are defined at every grid point in the u and v directions. The sources are defined by a linear combination of the first and second derivatives of one or more monitor functions, ϕ_n, in each parametric direction as shown in equations (12a) and (12b):

$$\sigma^u_{k\ell} = \sum_{n=1}^{M}\left\{w^u_{1n}\left|\frac{\partial\phi_n}{\partial u}\right| + w^u_{2n}\left|\frac{\partial^2\phi_n}{\partial u^2}\right|\right\} \tag{12a}$$

$$\sigma^v_{k\ell} = \sum_{n=1}^{N}\left\{w^v_{1n}\left|\frac{\partial\phi_n}{\partial v}\right| + w^v_{2n}\left|\frac{\partial^2\phi_n}{\partial v^2}\right|\right\} \tag{12b}$$

where k and ℓ are the indices of the grid point where the source is located. The w's are weighting parameters that allow the user to control the contribution from each term. The monitor functions ϕ can be any flow property (i.e., density, pressure, Mach number, etc.).

The control net in the parametric domain (u,v) is now modified using equations (13):

$$u'_{ij} = u_{ij} + \sum_{k=i-ic}^{i+ic}\sum_{\ell=j-jc}^{j+jc} K^u_{ijk\ell}\sigma^u_{k\ell} \tag{13a}$$

$$v'_{ij} = v_{ij} + \sum_{k=i-ic}^{i+ic}\sum_{\ell=j-jc}^{j+jc} K^v_{ijk\ell}\sigma^v_{k\ell} \tag{13b}$$

where $K_{ijk\ell}$ are the influence coefficients. The influence coefficient controls the effects of a source at a grid location (k,ℓ) on a control point (i, j). The (u',v') domain is normalized to go from zero to one. The modified new control net is shown in figure 4(d). The computation of the summation in equations (13) can be very costly if the influence coefficient K has to control the effects of the source on all grid points instead of the control points. This will be evident in a sample interactive case as the number of control points is significantly less than grid points. The coefficients are defined by equations 14(a) and 14(b):

$$K^u_{ijk\ell} = \frac{u_{ij} - u_{k\ell}}{d}\exp(-\alpha_u d) \tag{14a}$$

$$K^v_{ijk\ell} = \frac{v_{ij} - v_{k\ell}}{d}\exp(-\alpha_v d) \tag{14b}$$

where

$$d = \sqrt{\left(u_{ij} - u_{k\ell}\right)^2 + \left(v_{ij} - v_{k\ell}\right)^2}$$

and the α's are decay parameters for each of the parametric directions. The degree of adaptation is controlled by the user's input of the w's and α's.

In order to improve computational efficiency, cutoff distances have been implemented for the summations in equations (13). The cutoff distances ic and jc are defined so that the summations include sources located in the region where their influence coefficients are greater than the user-specified input, β. The input for this cutoff parameter β ranges between zero and one. The summation includes all points when $\beta = 0$ and only one point when $\beta = 1$.

Also, the sources are mirrored across each boundary to improve the orthogonality of the adapted control net to the boundary. Orthogonality of the control net lines can also be enforced by making the control net orthogonal to the boundary as described in reference 6.

The new control net points in the (u´,v´) domain have been repelled by the strong sources. This can be seen in figure 4(d). Now a new uniform control net is mapped back to the (s,t) domain using the mapping from the (u´,v´) domain to the (u,v) domain. This results in a clustering of points near the strong sources in the original parametric domain as shown in figure 4(e). At this point the new control net points are used to calculate a new parametric grid which is then mapped into the physical domain with the control net. The adapted control net in the physical domain is shown in figure 4(f).

5. Interactive Controls

A family of grid generation programs, called TURBO, is being developed at the Lewis Research Center. One member of TURBO is a menu-driven interactive grid generation code TURBO-I, which uses the control point form discussed in section 2. Most interactive framework and features of TURBO-AD came from TURBO-I (refs. 5 and 6) and are not presented in this report. Additional interactive features have been implemented in TURBO-AD. An example of this is the real time grid modification as a control point or group of control points is moved. Another example is the global change of control point distribution as a control point is moved using the workstation mouse.

CFD solutions often contain both strong and weak flow features. The strong features are easily captured with grid adaptation. However, the weak features are typically much more difficult to capture and many times are completely missed. An example of this would be an oblique shock with a reflection. The initial oblique shock would be the dominant feature which could easily be captured. However, the weaker reflected shock is still an integral part of the flow domain. Therefore, a source filter was implemented that would allow the user to capture weaker flow features if desired. Once the sources have been calculated, the user can look at the color contours of the source distribution. The strongest sources show up in red while the weakest sources are shown in blue with a smooth transition in between. If after an initial adaptation of the control net the weaker features are not adequately captured, a source filter can be used. In order to amplify the effects of weaker sources, the source filter elevates a source strength to σ_{max} when it is above a user specified value and eliminates it otherwise:

$$\sigma^u = \frac{1}{2}\sigma^u_{max}\left[1 + sign\left(\sigma^u - e^u\sigma^u_{max}\right)\right] \quad (15a)$$

$$\sigma^v = \frac{1}{2}\sigma^v_{max}\left[1 + sign\left(\sigma^v - e^v\sigma^v_{max}\right)\right] \quad (15b)$$

In equations (15), the user specifies the cutoff values e^u and e^v for the u and v directions, respectively. A demonstration of this source filtering is shown in the example of section 6.

A second filter was implemented in the source calculation stage that helps filter noise from the source distribution. It applies equation (16) to the source distribution using e^3 from the input file:

$$\sigma = \sigma\left(\frac{\sigma}{\sigma_{max}}\right)^{e^3} \quad (16)$$

When e^3 is approximately zero, no filtering is performed.

Another feature that was implemented contends with boundary grid integrity at boundary discontinuities. This option allows users to select points which are key features of the grid, such as boundary discontinuities, and ensure that a grid point is at that location before exiting from TURBO-AD. The process entails two steps. The first step takes places as soon as the user starts TURBO-AD. At this point the user selects "Mark Critical Pnts" from the "FIX CRITICAL POINTS" submenu. This allows the user to pick one or more of these anchor points interactively. If an incorrect point is accidently selected, the user can deselect the point by picking it again. The user should exit this operation when all critical points have been selected. The user should proceed with the grid modifications as usual. At the end of the TURBO-AD session the user should return to the submenu "FIX CRITICAL POINTS" and select "Adjust Grd to Critical Pnts". This option will determine the nearest new grid points to the previously selected locations and then shift these points to the necessary locations. The points around the shifted points are also shifted with an effect that dies out away from the prime point. The example of section 6 requires that this option be used.

6. Interactive Solution-Adaptive Grid Generation, Sample Case

The following example demonstrates many of the features available in TURBO-AD. The case to be considered is a ramp/expansion with an inflow Mach number of 2.0. The initial control net, grid, and Mach number distribution can be seen in figure 5. The final solution-adapted control net, grid, and new Mach number distribution is shown in figure 15. The new grid was developed through a sequence of steps that are explained in greater detail in the following paragraphs. The flow chart provided in appendix A contains the major steps of this example.

The following is an example of the events used to adapt the initial grid shown in figure 5(b) to the Mach number distribution also shown in figure 5(c). This sample case demonstrates many of the important features available in TURBO-AD. The initial grid, initial flow solution, and grid adaptation parameter files that are used in this example are available for TURBO-AD

users. TURBO-AD uses the mouse for the primary input. The right button either exits a current operation or displays the current menu. The menu selection is accomplished by highlighting the appropriate choice and then releasing the right button. The left and center mouse buttons are typically used for controls within the operations.

TURBO-AD can be executed by typing 'turboad.' The program then ask the user to type in the grid file name followed by the solution file name. The example files used here are called 'grd2.2d' and 'sol2.2d.' The solution file can contain one to five scalar functions. In this case, the file 'sol2.2d' contains density, pressure, temperature, Mach number, and entropy, in that order. The solution file is optional. If a 'return' is entered at the request for a solution file, the user has access to many of the extensive interactive grid modification features discussed in reference 6 about TURBO-I.

Once TURBO-AD has been started, it will create the parametric mapping in memory and then display the initial grid and control net. The user can then bring up the home menu by pressing the right mouse button and holding it down. This menu is the top level or main menu. The user can select "MODIFY VIEW" which will take the user to the view control menu. Then by pressing the right mouse and holding it down, the user will see the "MODIFY VIEW" menu choices. The input function distribution can be displayed by selecting "Contour Function." This is a pull down menu, so the user should move the mouse to the right side of the selection. This will bring down a listing of functions that can be contoured. By selecting the fourth function the user can see the Mach number contour for the example case. It should be noted that the first function in the user's solution file is "function 1," the second is "function 2," etc. The "Scroll View" option allows the user to look at the control net alone, the grid alone, or many combinations. For this example, the user should see the control net, grid, and Mach number distribution if "Scroll View" is selected several times. The initial control net, grid, and Mach number distribution that should be observed is shown in figures 5(a) to 5(c).

The example problem at hand has two critical points that define the point of compression and the point of expansion. These point locations should be maintained. Therefore, the user should push the right mouse button to bring up the "MODIFY VIEW" menu again and select "Zoom & Translate." The view should be zoomed and translated such that these two critical points are easily located and selected. This can be done by using the left mouse button to translate the view and the center button to zoom the view. Instructions for this should show up on the TURBO-AD screen. Once the critical points are clearly visible, the user should use the right button to exit "Zoom & Translate" and then select the "HOME" option from the menu. This takes the user to the main menu; when pressing the right mouse button, the user will see the main menu options. The user should select "FIX CRITICAL POINTS." Then the user should display this menu and select "MARK CRITICAL POINTS." This option will instruct the user to select the necessary critical points with the left mouse button. If the wrong point is selected, merely click on that point again to deselect the point. After selecting the point defining the compression and the point at the expansion the user should see something similar to figure 5(d). The user can select any number of critical points. The points can also be within the interior of the grid. The user should use the right mouse button to exit this option and then return to the HOME menu again using the appropriate menu selections. It should be noted that a point may not always be at these critical points during the session. However, before the user exits, the menu option "Adjust Grd to Critical Pnts" should be selected. This option will then make the necessary adjustments to the grid. The results of this will be demonstrated near the completion of this example.

The user is now ready to zoom back to the full view, using the "Zoom & Translate" feature, to prepare for the adaptation. The "ADAPTATION CONTROL" menu option can now be selected from the main menu or the "MODIFY VIEW" menu. Under the "ADAPTATION CONTROL" menu the user should use the mouse to pull down the "Active Functions" menu. This menu controls which functions are used for adaptation. The default is for all functions to be turned off. The user can toggle the functions on or off by selecting the appropriate one. For this example the fourth function is selected. Please note that there is no correlation between the function being viewed and the functions selected under "Active Functions" for the CN adaptation.

The 'adapt.par' file contains the user definable parameters w's and α's shown in equations (12) and (14). The formats of the input, output, and parameter files are discussed in appendix B. The file 'adapt1.par' should be copied to 'adapt.par' for the first part of the exercise. Then, 'adapt2.par' will be used for the latter part. The 'adapt1.par' file is included in appendix B. The user is now ready to calculate the sources discussed in equations (12a) and (12b). This can be done by selecting the menu option "Calculate Source Terms." There is no obvious sign that this option is finished. Therefore, the user should attempt to bring up the menus again using the right mouse button. When the calculation of the sources is finished, the user will succeed in bringing up the menu. The calculation of the sources has now produced two new functions in the users list of functions. This was done so the user could view the source distribution using the built in color contouring available under "MODIFY VIEW." If the user contours the sixth function for this example case, using the 'adapt.par' file, the user should see the source distribution in the u direction shown in figure 6. The blue colors show the lower source strength and the red show the higher source strength. The locations where high values exist is where the adapted control net lines should cluster. The source distribution in the v direction has been set to zero using the input file in this example. The user can now return to the adaptation menus using the appropriate selections.

The adaptation process can now be started. To do this the user should select "Adapt CN" from the "ADAPTATION CONTROL" menu. This operation will take about 20 to 90

seconds depending on which IRIS 4D series workstation is being used. The completion of this operation will be obvious if the control net is displayed. The user will see the new adapted control net pop up. The new control net should look like the one shown in figure 7. The user now has two choices: (1) Is the control net acceptable, or (2) should the user select "restore CN and Grid" to bring back the initial control net? The control net that now exists will capture the strong shock but may not capture the reflected shock adequately. If a user selects "ReCalculate Grid," TURBO-AD computes the new grid shown in figure 8 from the adapted control net shown in figure 7.

There are two options present to deal with the weaker reflected shock. The first is to continue at this point and recalculate the grid with the new control net using the "ReCalculate Grid" option as above and then use the "Redefine BLOCK" option under the HOME menu to work only on the area of the grid where the weaker shock is located. This will cut off the influence of the strong shock on the weaker one allowing it to be better captured. The second option is to use filtering as discussed in section 5. Both approaches are now illustrated.

The ability to subdivide the domain that the user is working on is a very useful option that can be used in many places. This option will be discussed based on its use in controlling adaptation. The menu option "Redefine BLOCK" is located under the HOME menu. The user can now select this option. TURBO-AD will instruct the user to select the (i_{min}, j_{min}) point using the left mouse button and the (i_{max}, j_{max}) point using the center button. The left button should be used to select a point near the expansion corner. The user can determine what region is selected by the color changes of the grid. The region can be changed over and over while in this option. To exit, the user should press the right mouse button. A new control net will be created and attached to the new local region as shown in figure 9(a). At this time the user should open another window and change the values of w_1^u from 0.006 to 0.003 and w_1^v from 0 to 0.003 for function 4 in the 'adapt.par' to adapt the control net in both u and v directions. Then the user should return to the adaptation menus and select "Calculate Sources." This will only calculate sources in the new region as shown in figures 9(b) and (c) for the u and v directions, respectively. The user can then use "Adapt CN" and "ReCalculate Grid" to adapt to the weaker shock. The process should be repeated two times starting with the calculation of the sources. The user should note that there are discontinuities along the local block boundaries. This will be smoothed out when the user returns to "Redefine BLOCK" and chooses the entire grid and does a grid calculation. The resulting control net and grid are compared with those before the subblock adaptation in figures 10 and 11. Improvements can now be seen in the new grid. Further improvements can still be made by using local control point manipulations as discussed in reference 6. One of the nice things about interactive environments is the ability to look at the grid as it changes.

TURBO-AD can also capture both strong and weak flow features by filtering the control sources without using the subblock operation as illustrated in the previous paragraph. In order to see the effect of the filtering clearly, the user should quit TURBO-AD using the "QUIT" option in the main menu and restart it. This allows the user to return to the initial grid and control net. The user should then proceed through the above procedure up to the first adaptation. At this point the user should pick the critical points and the function 4 (Mach number) to be used in the adaptation. The adaptation parameters are a little different for this exercise. Therefore, the user should go to another window and copy 'adapt2b.par' to 'adapt.par'. The user should proceed to "Calculate Sources." The filtering option "Filter Sources" can now be selected from the adaptation menu. This filtered source distribution can then be viewed and should look like figure 12. The sources can now be considered on or off as discussed in section 5. There are no intermediate values. The user can now select "Adapt CN." When this operation is finished the control net should be adapted more strongly near the weak shock. The grid should now be calculated. The resulting grid and control net are shown in figure 13.

The adapted grid shown in figure 13(b) can be further modified by using explicit local or global control features of TURBO-AD. The user should go to the "MODIFY CONTROL NET" menu. Under this menu the user should pull down the "Choose & Move CP" option and select "Choose Active CP." This option will instruct the user to move the red highlighted control net lines to a desired control point (CP) using the left and center keys of the mouse button. The user has the option to select "Toggle real time Move reg." (ion) to let the grid be updated as the highlighted CP is moved. The user can then pick "Move Chosen CP and region" from the menu entry to translate the chosen CP and to redistribute the CP's nearby. The mouse pointer will now pop up at the highlighted intersection. Then the highlighted CP's will begin to follow the mouse pointer. After some experimenting with this option, the user should be able to obtain desired fine tuning of the grid. For the current sample case, the user should be able to obtain a control net similar to the one shown in figure 14 which results in a grid without excessive skewness below the reflected shock.

The final step is to recapture the sharp corner points in the grid. The user may want to zoom into this region to watch the process happen. To accomplish this, the user should return to the main menu and select "FIX CRITICAL POINTS." Under this menu the user should select "Adjust Grd to Critical Pnts." The movement made to recapture the grid points can be seen if watched carefully. At this point the user has an adapted grid that should be written out using TURBO-AD format or Plot3d format. These options may be found in a pulldown menu under the main menu. The TURBO-AD option writes a file called 'gridx.2d', and the "Plot3d" option writes a file called 'gridx.fmt', where x is the number of times you have selected the respective option. The "Plot3d" option will also write out a file called 'funx.2d' which contains the solution distribution interpolated for the modified grid.

The user can now see the new adapted control net, grid, and flow solution shown in figure 15. The new solution was

obtained using the adapted grid. The strong and weak shocks have both become more crisp and discernable. There are many features in TURBO-AD that have not been used in this example. The other features have been discussed in reference 6 to some detail. In a short period of time a user can become proficient at TURBO-AD because of its interactive nature.

7. Concluding Remarks

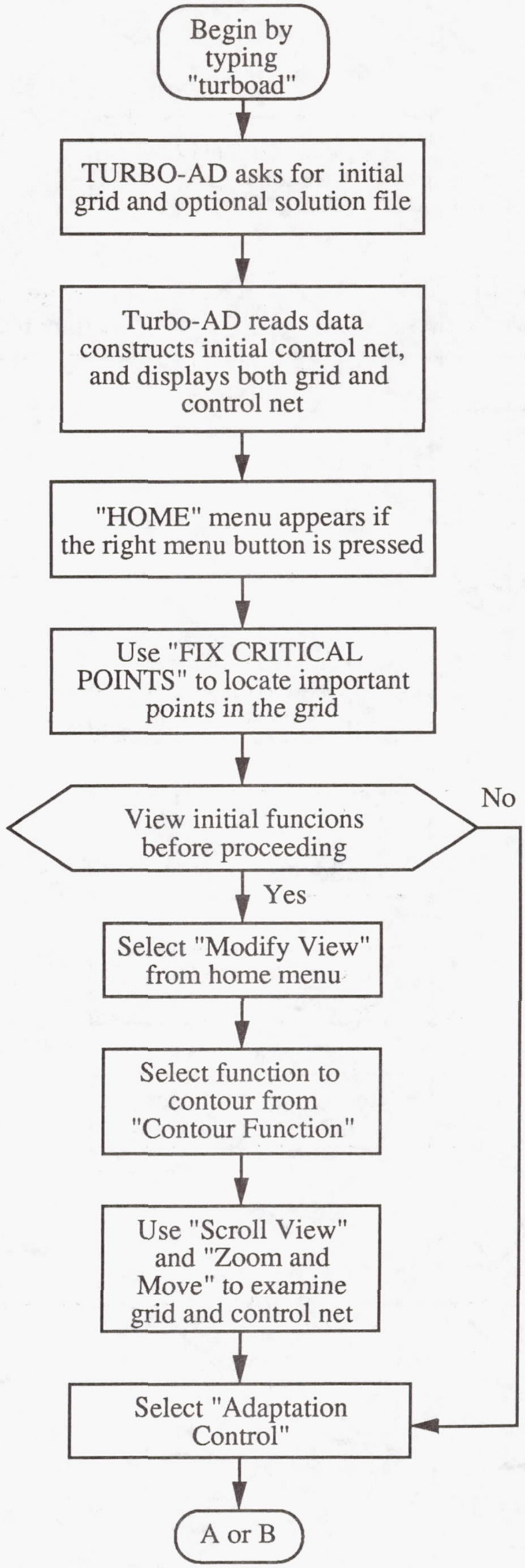
Two technologies are combined to develop an interactive solution-adaptive grid generation program, TURBO-AD. One is the solution-adaptation technique that uses the parametric mapping with control sources, that are derived from the flow solutions. The other is the control point form of algebraic grid generation technique that provides precise local control. The control point form technique is efficient as an algebraic method. Significant computational savings are attained by adapting the control net to solutions instead of the usual direct grid adaptation to solutions. Further savings in the computational time are achieved in TURBO-AD by cutting off the computation of the influence function K at grid points where K decays below a user specified value.

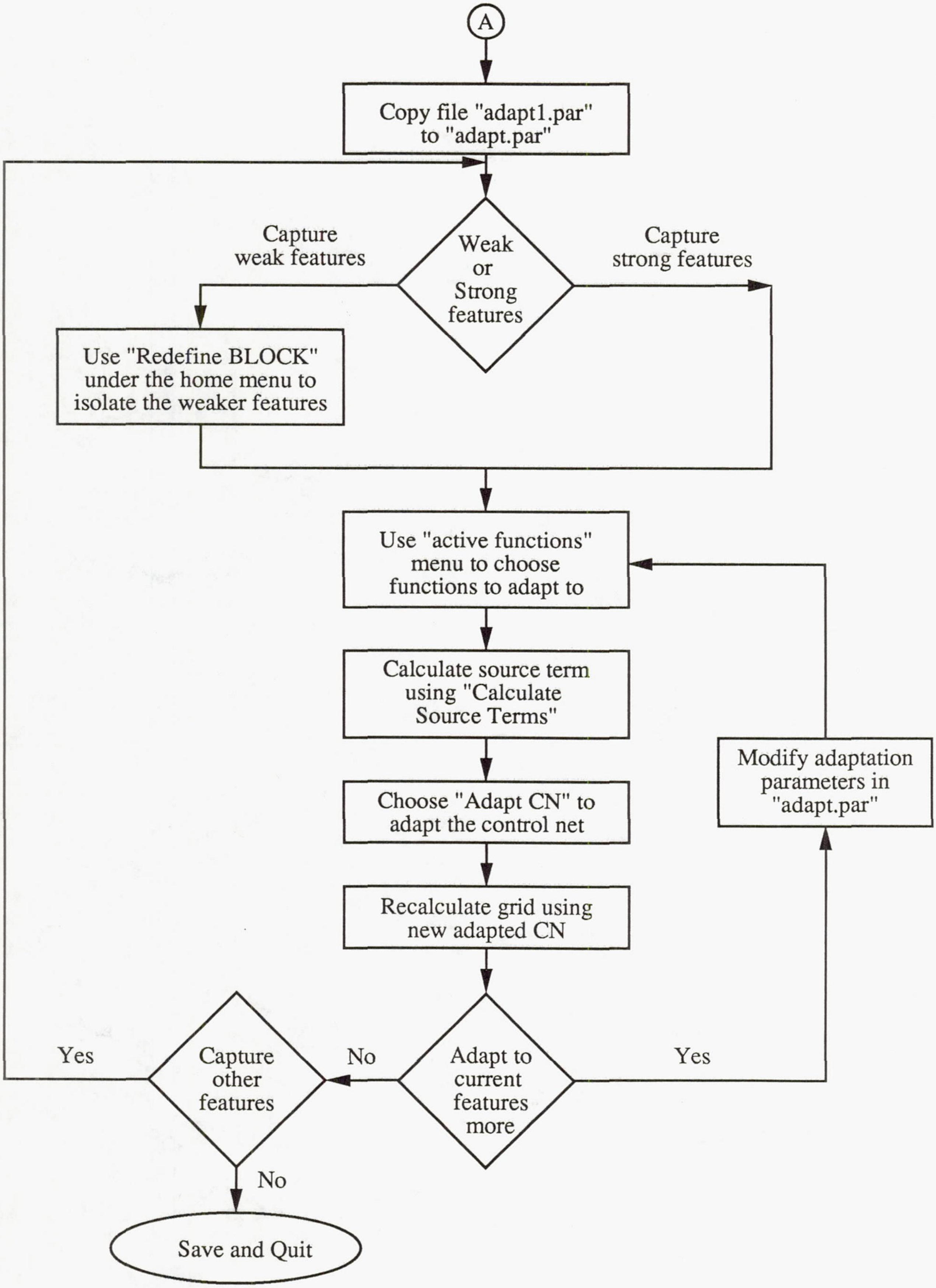
TUROB-AD is being developed on the IRIS 4-D series workstations. It can display the grid, solution contour, and the control net separately or can overlay one over the other for examination. With the improvement in computational speed, two-dimensional solution-adaptive grid generation is now a viable option on graphics workstations. As shown with an illustration, TURBO-AD has global grid control over the entire domain, control over subregions, as well as local grid control capability.

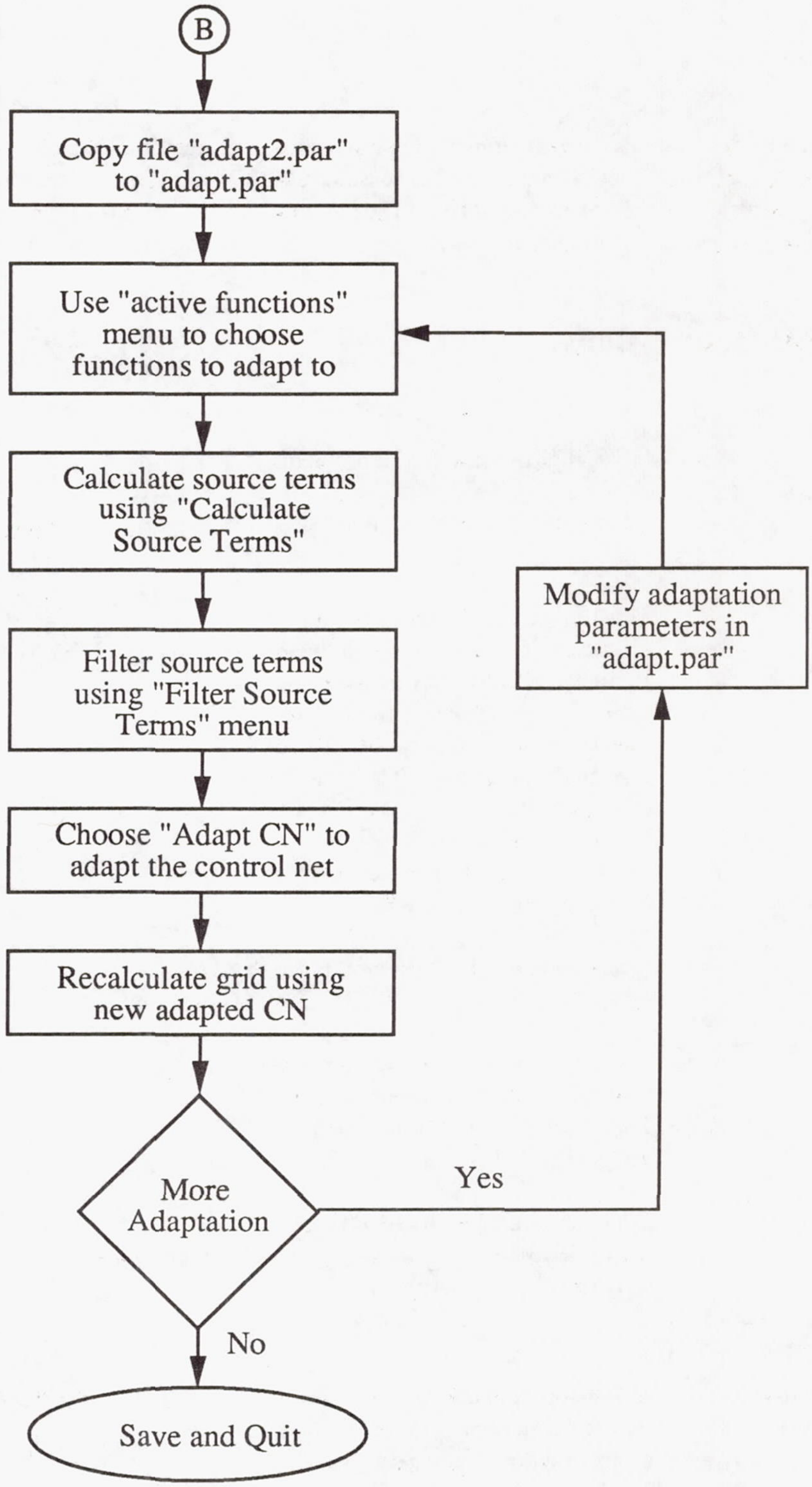
TURBO-AD has had limited testing. Further testing and enhancements are needed to improve robustness and efficiency of the code. Also needed are guidelines for the selection of adaptation parameters and integration of a solver.

Appendix A

Flow Chart For Sample Case




A
Copy file "adapt1.par" to "adapt.par"
Capture weak features
Weak or Strong features
Capture strong features
Use "Redefine BLOCK" under the home menu to isolate the weaker features
Use "active functions" menu to choose functions to adapt to
Calculate source term using "Calculate Source Terms"
Modify adaptation parameters in "adapt.par"
Choose "Adapt CN" to adapt the control net
Recalculate grid using new adapted CN
Yes
Capture other features
No
Adapt to current features more
Yes
No
Save and Quit


B
Copy file "adapt2.par" to "adapt.par"
Use "active functions" menu to choose functions to adapt to
Calculate source terms using "Calculate Source Terms"
Filter source terms using "Filter Source Terms" menu
Choose "Adapt CN" to adapt the control net
Recalculate grid using new adapted CN
More Adaptation
Yes
Modify adaptation parameters in "adapt.par"
No
Save and Quit

The following descriptions cover the input and output files particular to TURBO-AD. Plot3d ascii formats are assumed to be widely recognized and are not discussed here. The numerical values are read using '*' format statements.

Input File Descriptions

Grid File

Line 1	==>	Header line used to describe problem.
Line 2	==>	Description line to aid in modifying inputs.
Line 3	==>	This line actually contains the input information. All of these numbers are integers. The first variable is the number of grid lines in the i direction, the second is the number of grid lines in the j direction. The number of control net lines in the i and j directions are specified next in that order. The last two values are for compatibility with TURBO-I.
Line 4	==>	Description line to aid in modifying inputs.
Line 5	==>	This line contains two integers and two real numbers that are never used. They have been left in for compatibility with TURBO-I.
Additional lines	==>	Grid data using two implied loops, ((x(i,j),i=1,imax),j=1,jmax) ((y(i,j),i=1,imax),j=1,jmax).

Solution File

The first line of the solution file contains the number of points in the i and j directions and the number of functions to be read. These values can be in either floating or integer form. The first line is followed by all of the first function, then all of the second function, etc., up to five functions.

Adaptation Parameter File

The adaptation parameter file contains the parameters that control the degree of adaptation. This includes the w's in equation (12), σ's in equation (14) in section 4, and the filtering parameters. The following is a sample file followed by a brief description of each line. This is the 'adapt1.par' file used for the first part of the sample case in section 6.

αu	αv	e^3	β	e^u	e^v
5.0	5.0	0.0001	0.10	0.00015	0.01
wu1	wu2	wv1	wv2	for function 1	
0.003	0.003	0.0	0.0		
wu1	wu2	wv1	wv2	for function 2	
0.003	0.003	0.0	0.0		
wu1	wu2	wv1	wv2	for function 3	
0.003	0.003	0.0	0.0		
wu1	wu2	wv1	wv2	for function 4	
0.006	0.003	0.0	0.0		
wu1	wu2	wv1	wv2	for function 5	
0.003	0.003	0.0	0.0		

Line 1	==>	Description line to aid in modifing inputs.
Line 2	==>	This line contains the decay parameters, in equation (14), for the u and v directions, the e^3, e^u, and e^v filtering parameters in section 5, the cutoff value for the adaptation summation that was discussed in section 4.
Line 3	==>	Description line to aid in modifying inputs.
Line 4	==>	This line contains the w's discussed in section 4 for u and v directions.
Additional lines	==>	There should be two lines similar to lines 3 and 4 for each function read.

Output File Description

The output file from TURBO-AD is identical to the input file.

References

1. Lee, K.D.; Loellbach, J.M.; and Pierce, T.R.: Solution Adaptive Grid Generation Using a Parametric Mapping. Numerical Grid Generation in Computational Fluid Mechanics '88: Proceedings of the Second International Conference, S. Sengupta, et al., eds., Pineridge Press, Ltd, Swansea, Wales, 1988, pp. 455–464.

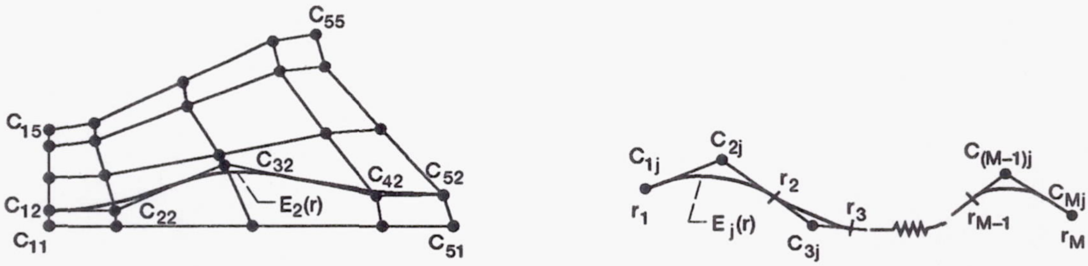
2. Lee, K.D.; and Loellbach, J.M.: Geometry-Adaptive Surface Grid Generation Using a Parametric Projection. J. Aircraft, vol. 26, Feb. 1989, pp. 162–167.

3. Lee, K.D.; Henderson, T.L.; and Choo, Y.K.: Grid Quality Improvement by a Grid Adaptation Technique. Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, A.S.-Arcilla, et al., eds., North-Holland, 1991, pp. 597–606.

4. Eiseman, P.R.: A Control Point Form of Algebraic Grid Generation. Int. J. Num. Methods in Fluids, vol. 8, Oct. 1988, pp. 1165–1181.

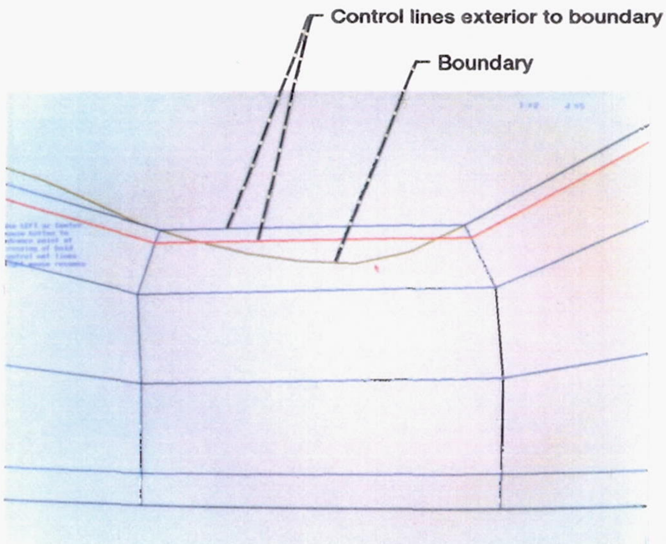
5. Eiseman, P.R.; Snyder, A.; Choo, Y.K.: Dynamics of Local Grid Manipulations for Internal Flow Problems, Computational Fluid Dynamics Symposium on Aeropropulsion, NASA CP–10045, 1990, pp. 30–26, 1990.

6. Choo, Y.K.; Miller, D.P.; and Reno, C.: Implementation of Control Point form of Algebraic Grid Generation Technique, Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, ed. by A.S.-Arcilla, et al. North-Holland, pp. 331–341, 1991.

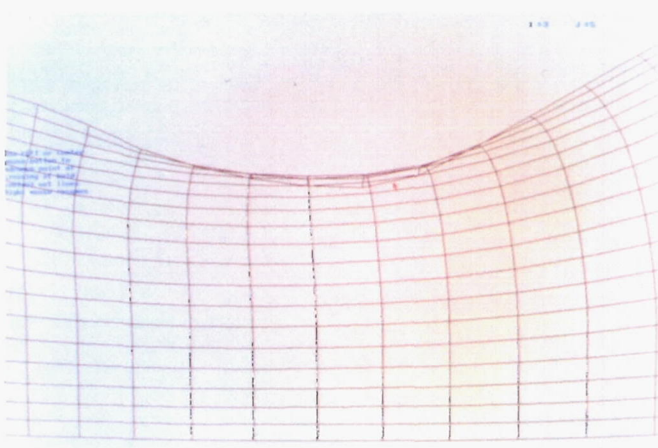


(a) Control net and a curve. (b) Construction of a curve.

Figure 1.—Construction of grid using control point form.



(a) Control net at concave region.



(b) Tangled grid.

Figure 2.—Concave region causing tangling.

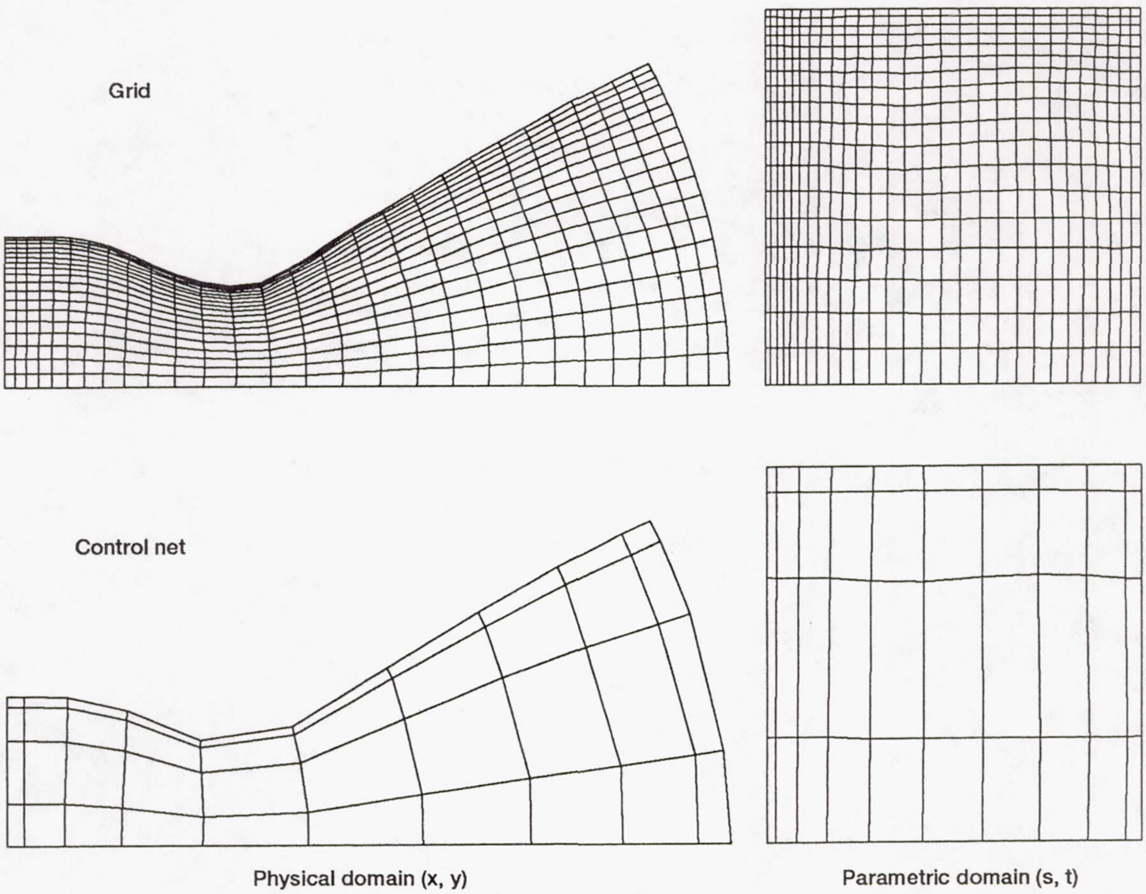
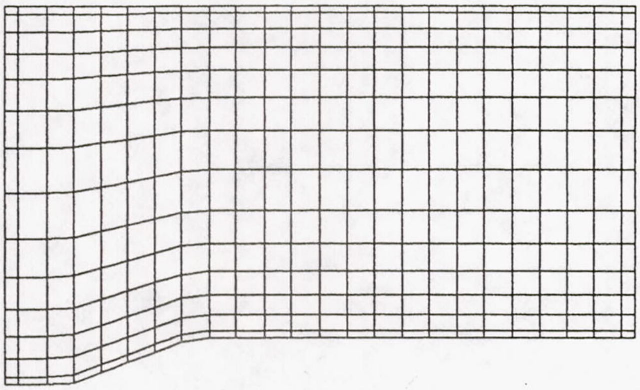
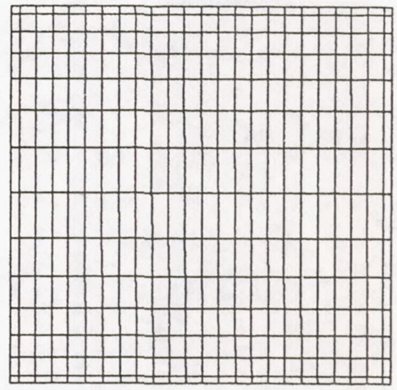


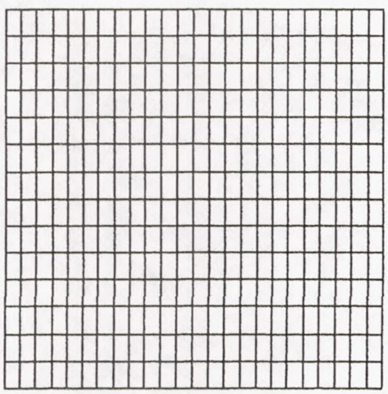
Figure 3.—Example of grid and control net in physical domain and parametric domain.



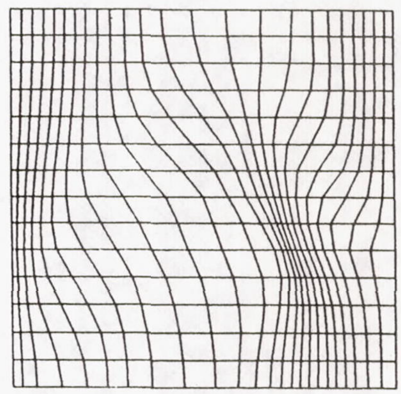
(a) Initial control net in physical space (x, y).



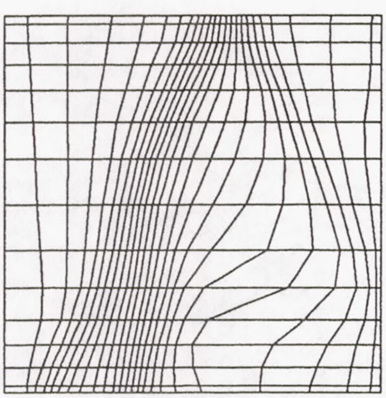
(b) Initial control net in parametric space (s, t).



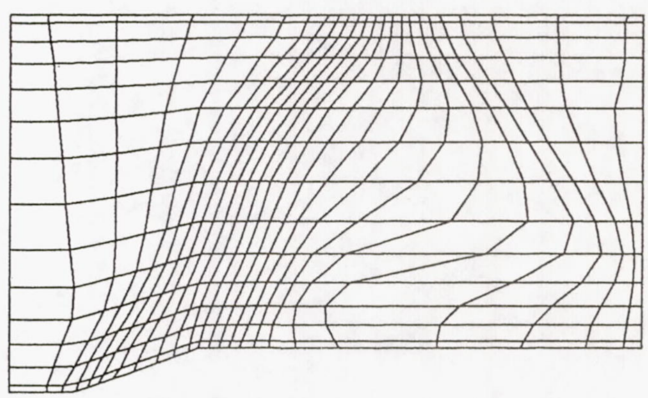
(c) Uniform control net in parametric space (u, v).



(d) Adapted control net in parametric space (u', v').

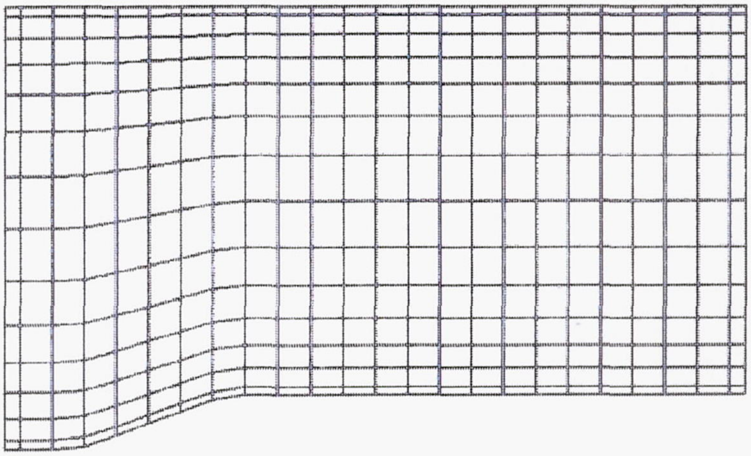


(e) Adapted control net in parametric space (s, t).

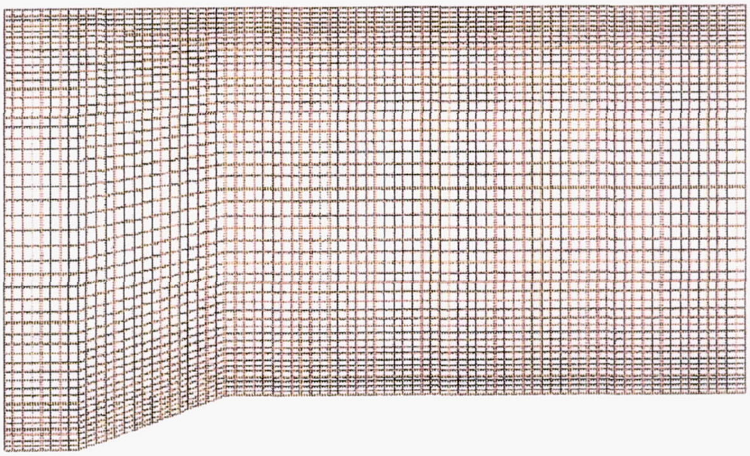


(f) Adapted control net in physical space (x, y).

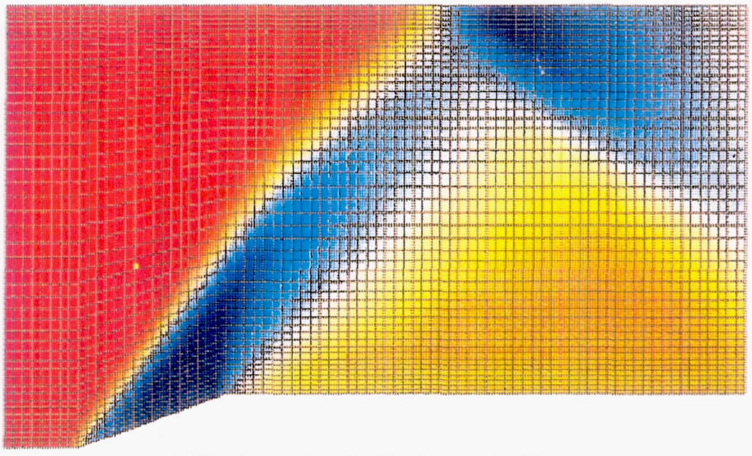
Figure 4.—Adaptation process for control net.



(a) Control net.

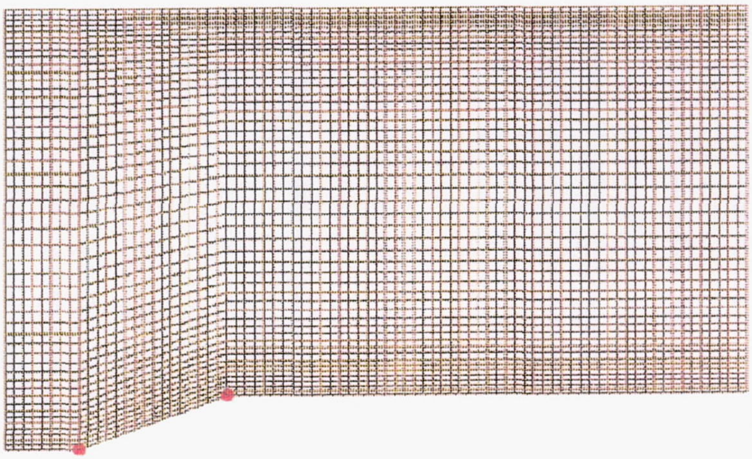


(b) Grid.



(c) Mach number distribution and grid.

Figure 5.—Initial control net, grid, Mach number distribution.



(d) View of critical points after selection.

Figure 5.—Concluded.

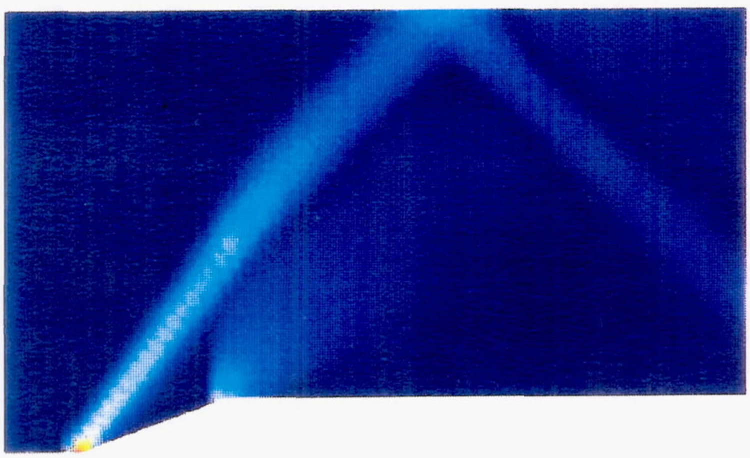
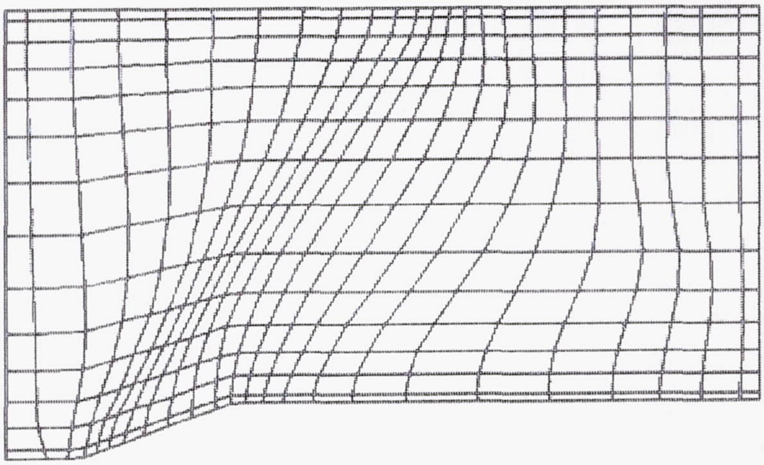
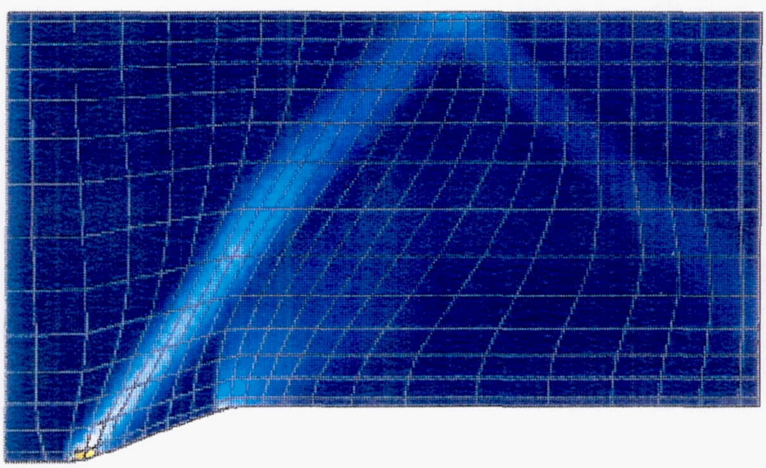


Figure 6.—Source distribution in u-direction without filtering.

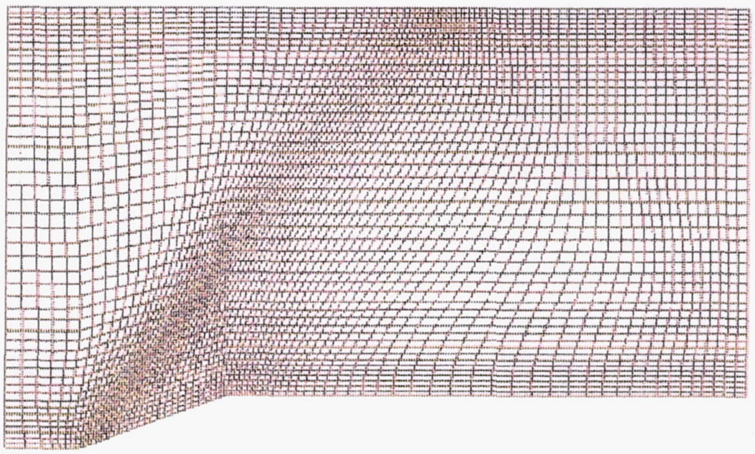


(a) Control net.

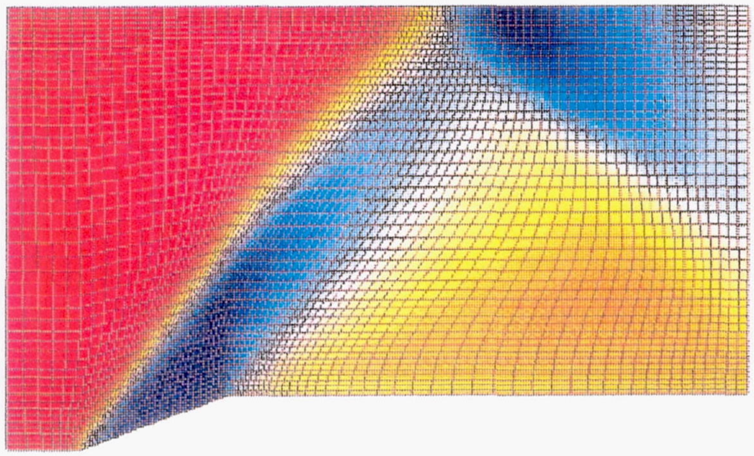


(b) Control net and source distribution.

Figure 7.—Adapted control net without filtering.

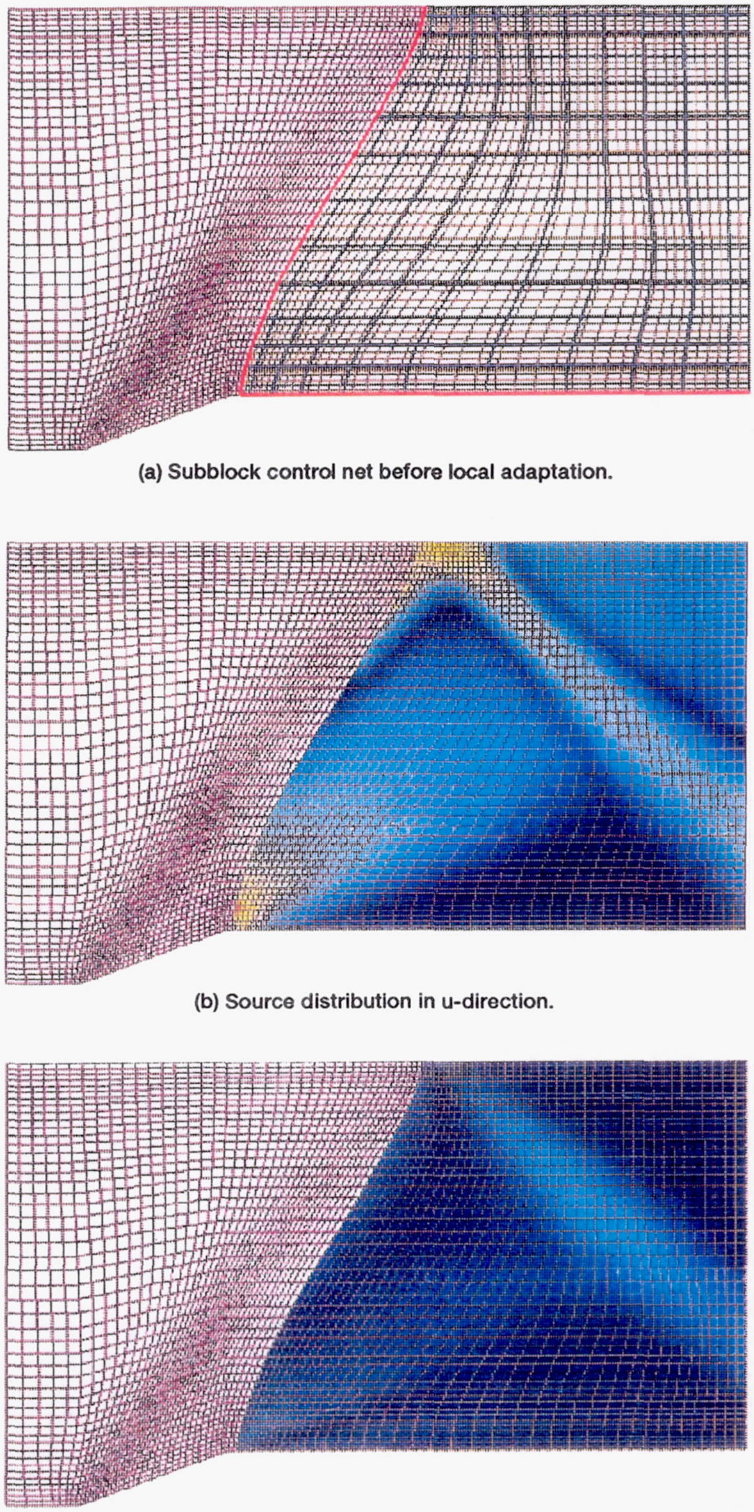


(a) Grid.



(b) Grid and initial Mach number distribution.

Figure 8.—Partially adapted grid.

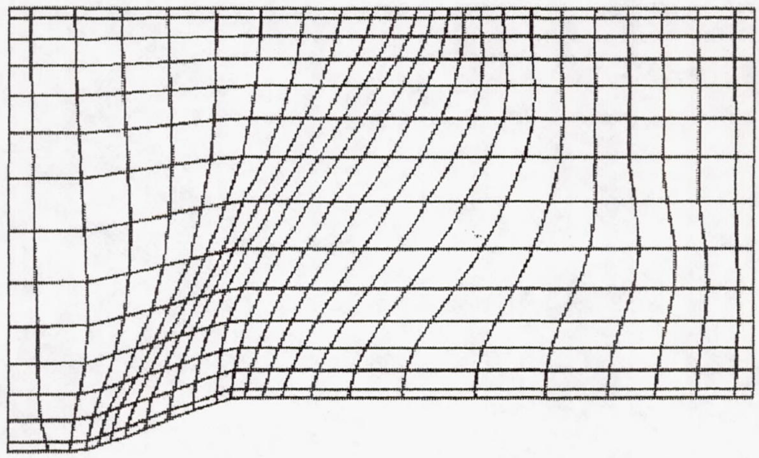


(a) Subblock control net before local adaptation.

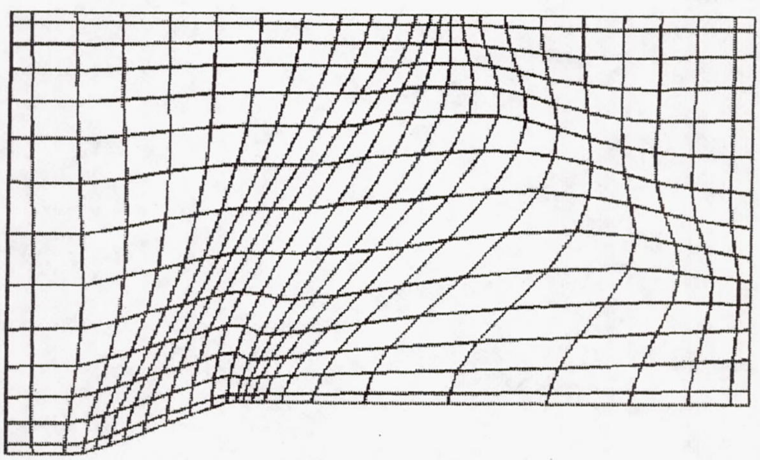
(b) Source distribution in u-direction.

(c) Source distribution in v-direction.

Figure 9.—Subblock used for local adaptation to resolve the weak reflected shock.

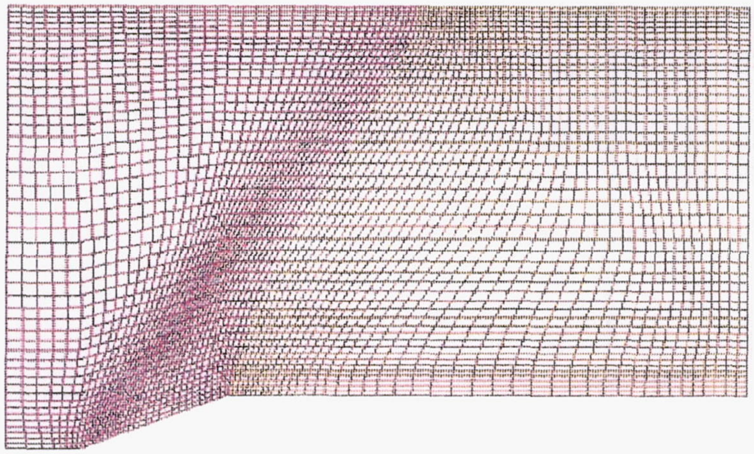


(a) Before subblock CN adaptation.

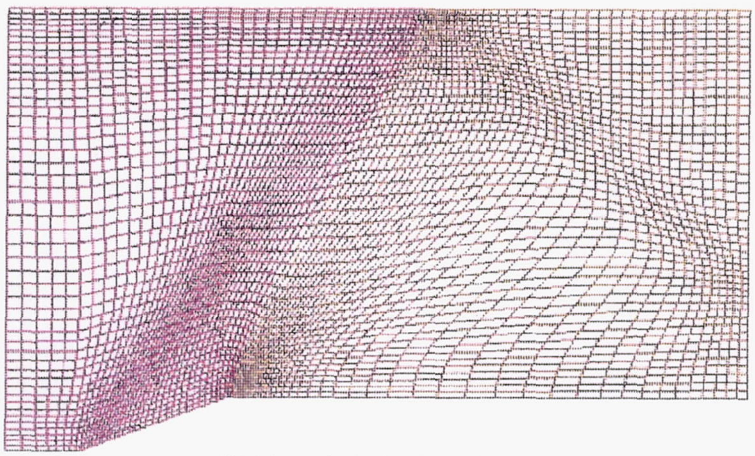


(b) After subblock CN adaptation.

Figure 10.—Comparison of control net.



(a) Before subblock adaptation.



(b) After subblock adaptation.

Figure 11.—Comparison of grid.

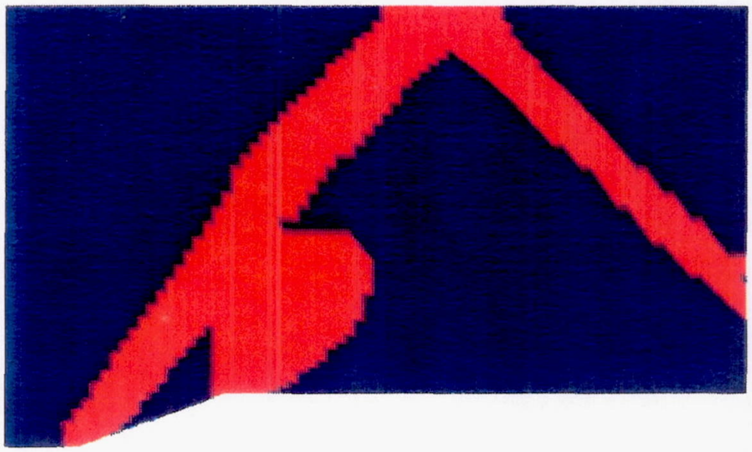
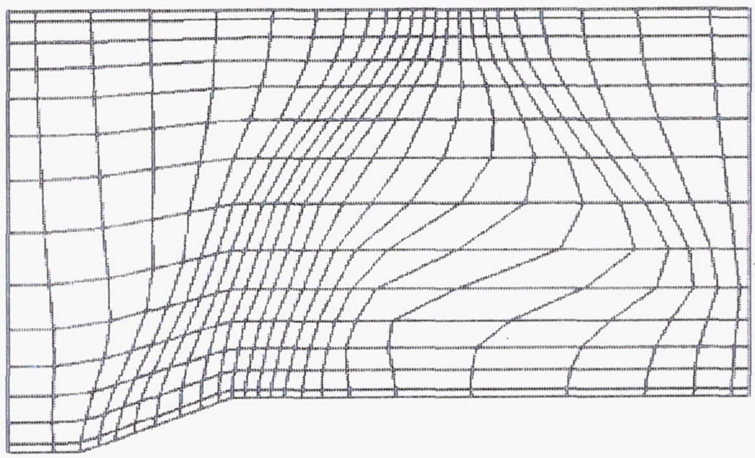
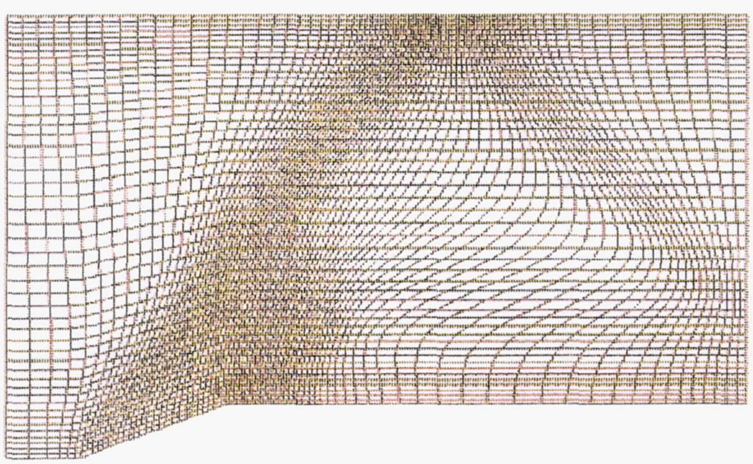


Figure 12.—Filtered source distribution.



(a) Control net.



(b) Grid.

Figure 13.—Adapted control net and grid from filtered source distribution.

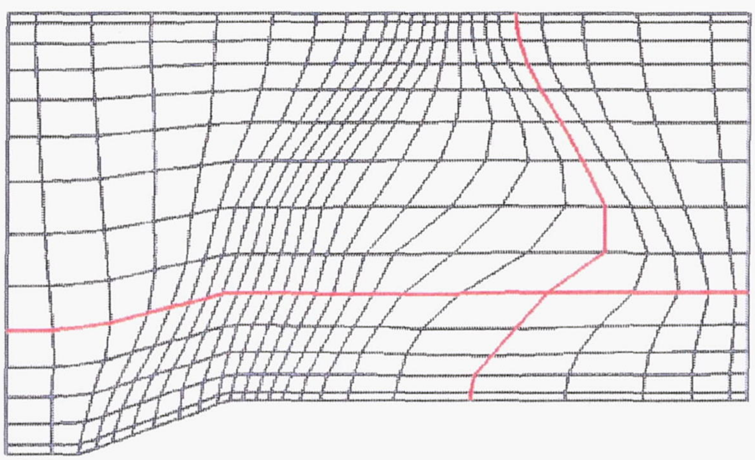
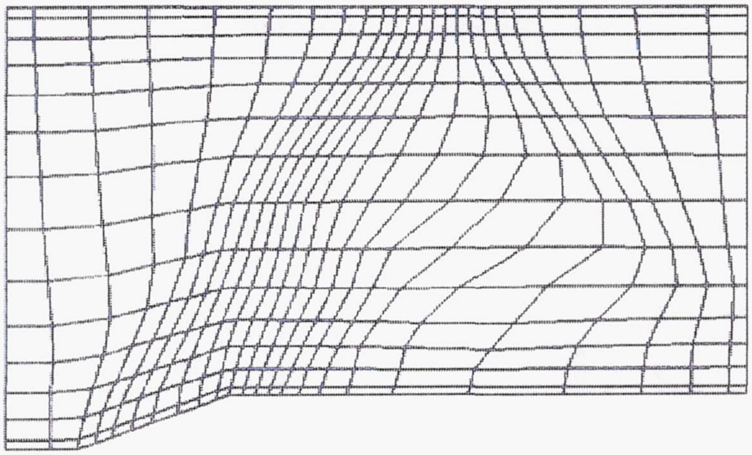
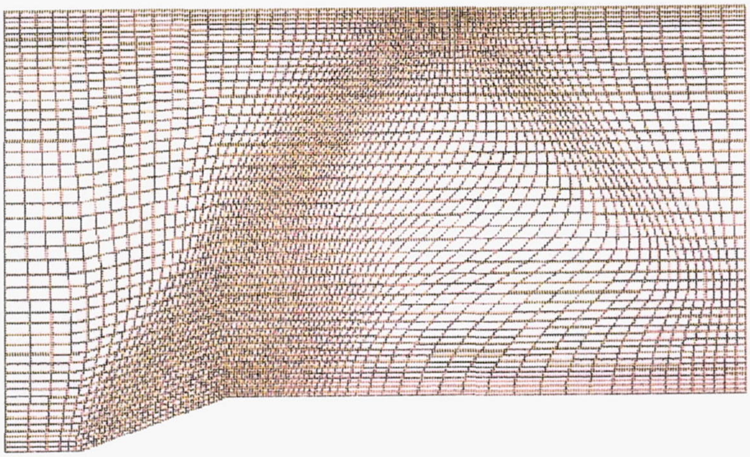


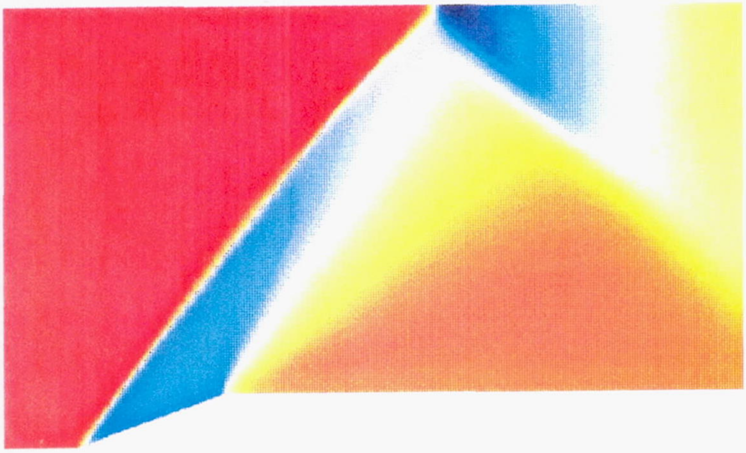
Figure 14.—Control net after explicit control.



(a) Control net.



(b) Grid.



(c) Mach number distribution after rerunning flow code.

Figure 15.—Final results.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1992	Technical Memorandum

4. TITLE AND SUBTITLE

Interactive Solution-Adaptive Grid Generation Procedure

5. FUNDING NUMBERS

WU–505–62–52

6. AUTHOR(S)

Todd L. Henderson, Yung K. Choo, and Ki D. Lee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E–6853

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM–105432

11. SUPPLEMENTARY NOTES

Todd L. Henderson, University of Illinois, Urbana, Illinois, 61801, and Summer Student Intern at NASA Lewis Research Center; Yung K. Choo, NASA Lewis Research Center; Ki D. Lee, University of Illinois, Urbana, Illinois, 61801. Responsible person, Yung K. Choo, (216) 433–5868.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Categories 2 and 7

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

TURBO-AD is an interactive solution-adaptive grid generation program under development. The program combines an interactive algebraic grid generation technique and a solution-adaptive grid generation technique into a single interactive solution-adaptive grid generation package. The control point form uses a sparse collection of control points to algebraically generate a field grid. This technique provides local grid control capability and is well suited to interactive work due to its speed and efficiency. A mapping from the physical domain to a parametric domain was used to improve difficulties that had been encountered near outwardly concave boundaries in the control point technique. Therefore, all grid modifications are performed on a unit square in the parametric domain, and the new adapted grid in the parametric domain is then mapped back to the physical domain. The grid adaptation is achievd by first adapting the control points to a numerical solution in the parametric domain using control sources obtained from flow properties. Then a new modified grid is generated from the adapted control net. This solution-adaptive grid generation process is efficient because the number of control points is much less than the number of grid points and the generation of a new grid from the adapted control net is an efficient algebraic process. TURBO-AD provides the user with both local and global grid controls.

14. SUBJECT TERMS

Computational grids; Computational fluid dynamic; Supersonic inlets

15. NUMBER OF PAGES
28

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	